1. A method of detecting vulnerabilities in source code comprising:
   analyzing variables in the source code and creating models therefrom in which each model specifies pre-determined characteristics about each variable;
   using the variable models to create models of arguments to routine calls in the source code; and
   using the argument models in conjunction with pre-specified criteria for the corresponding routine calls to determine whether the routine calls possess vulnerabilities as a consequence of the arguments and known routine behavior.

2. The method of claim 1 wherein the models specify the memory size of a variable.

3. The method of claim 1 wherein the models specify the data size of a variable.

4. The method of claim 1 wherein the models specify whether the variable is a null terminated string or not null terminated string for variables of string value type.

5. The method of claim 1 wherein the models specify the type of memory of the variable.

6. The method of claim 1 wherein the models specify the value of a string for variables that are of string value type.

7. The method of claim 1 wherein the models specify the origin of the data for a variable.

8. The method of claim 1 wherein the argument models specify characteristics of variable arguments.

9. The method of claim 1 wherein the argument models specify characteristics of expression arguments.

10. The method of claim 1 wherein the models are specified as lattices.

11. The method of claim 10 wherein the lattice values can include a value to represent no knowledge, a value to represent inconsistent knowledge, and a value to represent a refinement of knowledge.

12. The method of claim 11 wherein the value to represent a refinement of knowledge includes values to specify a range of specific values.

13. The method of claim 1 wherein the pre-specified criteria for the corresponding routine includes rules about the semantic behavior of the routine.

14. The method of claim 1 wherein the vulnerabilities are buffer overflows.

15. A method of detecting vulnerabilities in source code comprising:

analyzing source code to create models of arguments to routine calls in the source code, and

using the argument models in conjunction with pre-specified criteria for the corresponding routine calls to determine whether the routine calls possess vulnerabilities as a consequence of the arguments and the routine behavior.

16. A system for detecting vulnerabilities in source code comprising:

computer implemented logic for analyzing variables in the source code and creating models therefrom in which each model specifies pre-determined characteristics about each variable;

computer implemented logic for using the variable models to create models of arguments to routine calls in the source code; and

computer implemented logic for using the argument models in conjunction with pre-specified criteria for the corresponding routine calls to determine whether the routine calls possess vulnerabilities as a consequence of the arguments and known routine behavior.

17. The system of claim 20 wherein the computed implemented logic for using the argument models in conjunction with pre-specified criteria for the corresponding routine calls to determine whether the routine calls possess vulnerabilities as a

consequence of the arguments and known routine behavior includes a database specifying rules to detect vulnerabilities based on an analysis of the argument models.